

OpenXR™





Standardizing XR: OpenXR

Brent E. Insko, PhD
Lead VR Architect at Intel & OpenXR Working Group Chair

3GPP, VRIF & AIS Workshop on VR Ecosystem & Standards, April 2019

Agenda

- What is OpenXR?
- A Brief History of the Standard
- What are the Problems we are trying to Solve
- OpenXR Timeline of Development
- Provisional Release
- What's Next?
- Recap
- Call To Action
- Questions

OpenXR Provisional 0.90 Released!

The screenshot shows a press announcement page. At the top left is the Khronos Group logo with the tagline "CONNECTING SOFTWARE TO SILICON". A hamburger menu icon is in the top right. The main heading is "Khronos Releases OpenXR 0.90 Provisional Specification for High-performance Access to AR and VR Platforms and Devices". Below this is the date and location: "March 18, 2019 - 6:00 AM PT - Game Developer Conference, San Francisco". The text describes the release of the OpenXR 0.90 provisional specification, its purpose as a unifying, royalty-free standard for AR and VR, and where to find it. It also mentions that the specification can be found on the Khronos website and is released in provisional form to enable developers and implementers to provide feedback at the OpenXR forum. A paragraph follows, detailing that the OpenXR 0.90 provisional release specifies a cross-platform Application Programming Interface (API) enabling XR hardware platform vendors to expose the functionality of their runtime systems. By accessing a common set of objects and functions corresponding to application lifecycle, rendering, tracking, frame timing, and input, which are frustratingly different across existing vendor-specific APIs, software developers can run their applications across multiple XR systems with minimal porting effort—significantly reducing industry fragmentation. A final paragraph states that the Khronos OpenXR working group was formed in early 2017 with the support and participation of leading XR companies. Throughout the development of the specification,

Press announcement

The screenshot shows an updated landing page. At the top left is the Khronos Group logo with the tagline "CONNECTING SOFTWARE TO SILICON". A hamburger menu icon is in the top right. The background is a dark purple gradient. The main heading is "OpenXR™" in a large, white, stylized font. Below this is the sub-heading "UNIFYING REALITY". The text below reads: "OpenXR is a royalty-free, open standard that provides high-performance access to Augmented Reality (AR) and Virtual Reality (VR)—collectively known as XR—platforms and devices." Below this is the main heading "OpenXR 0.90 is Here!". The text below reads: "The OpenXR 0.90 provisional specification was released on March 18th 2019". At the bottom, there are two rows of buttons: "Press Release", "Provisional Specification", "Sample Code", "Launch Slides" in the first row, and "Reference Pages", "Overview Guide", "Merchandise" in the second row.

Updated landing page

What is OpenXR?

OpenXR is a royalty-free, open standard that provides high-performance access to Augmented Reality (AR) and Virtual Reality (VR)—collectively known as XR—platforms and devices.

A Brief History of OpenXR

- Among the first VR hardware available 2016















- Need applications...
 - Each platform provided an SDK to interface with the hardware
 - Each was different from the other

XR Ecosystem Fragmentation

- Increased development time and therefore cost.
- Increased validation overhead and therefore cost.
- Time and resources spent developing one title, impacts developers' ability to create more titles.

Major XR Runtimes

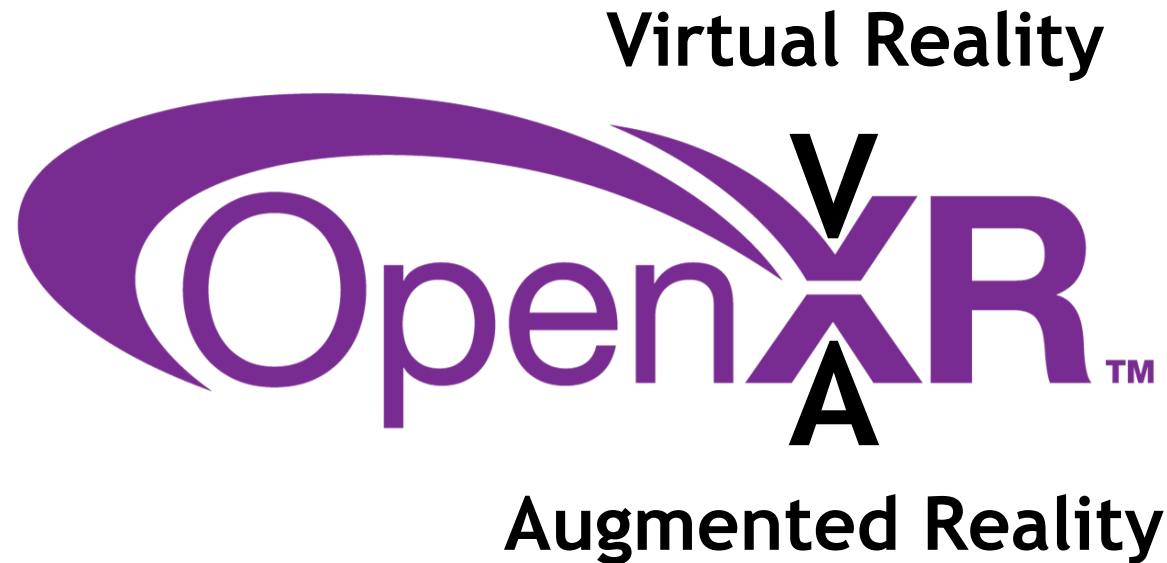
	Virtual Reality						Augmented Reality				Console VR
	PC			AIO	Mobile		AIO		Mobile		
	Oculus Rift	SteamVR	Mixed Reality	Oculus Go	Daydream	GearVR	Hololens	ML1	ARKit	ARCore	PSVR
Company	Facebook	Valve	Microsoft	Facebook	Google	Samsung Oculus	Microsoft	Magic Leap	Apple	Google	Sony
OS support		 									

VR Barriers to Adoption...

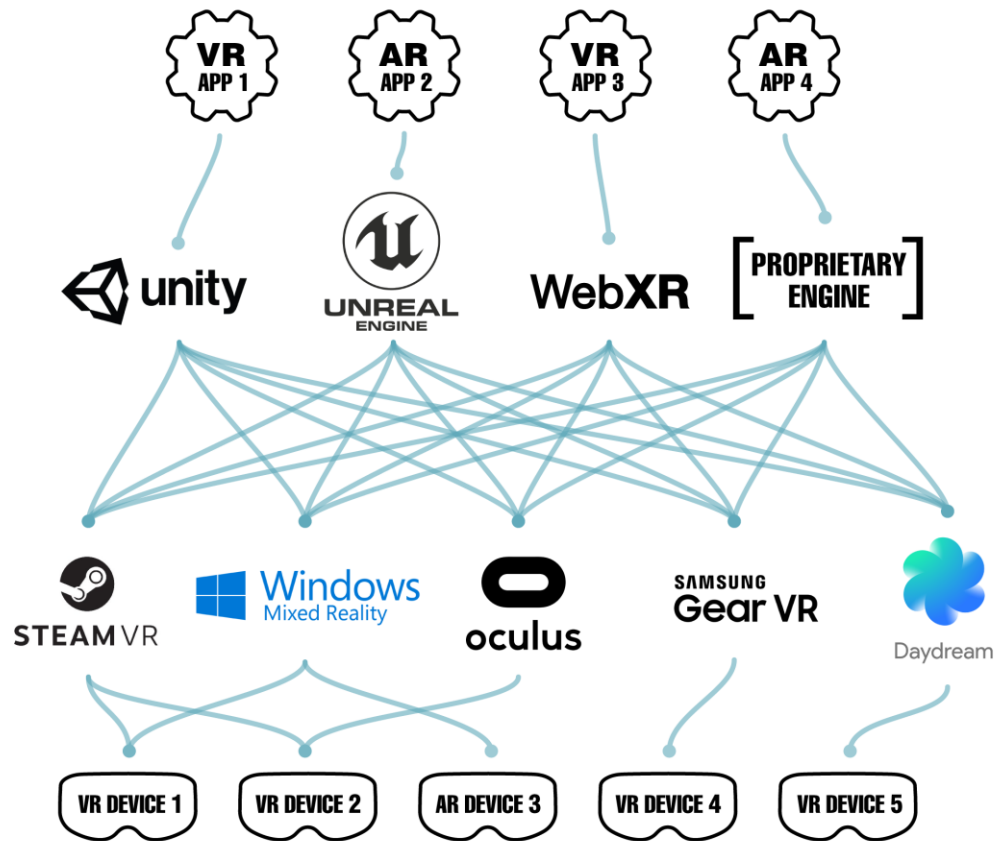
- **Content**
- **Comfort**
- **Cost**

OpenXR

- Recognizing the problem, several companies got together in late 2016 / early 2017 and formed the OpenXR working group in Khronos.



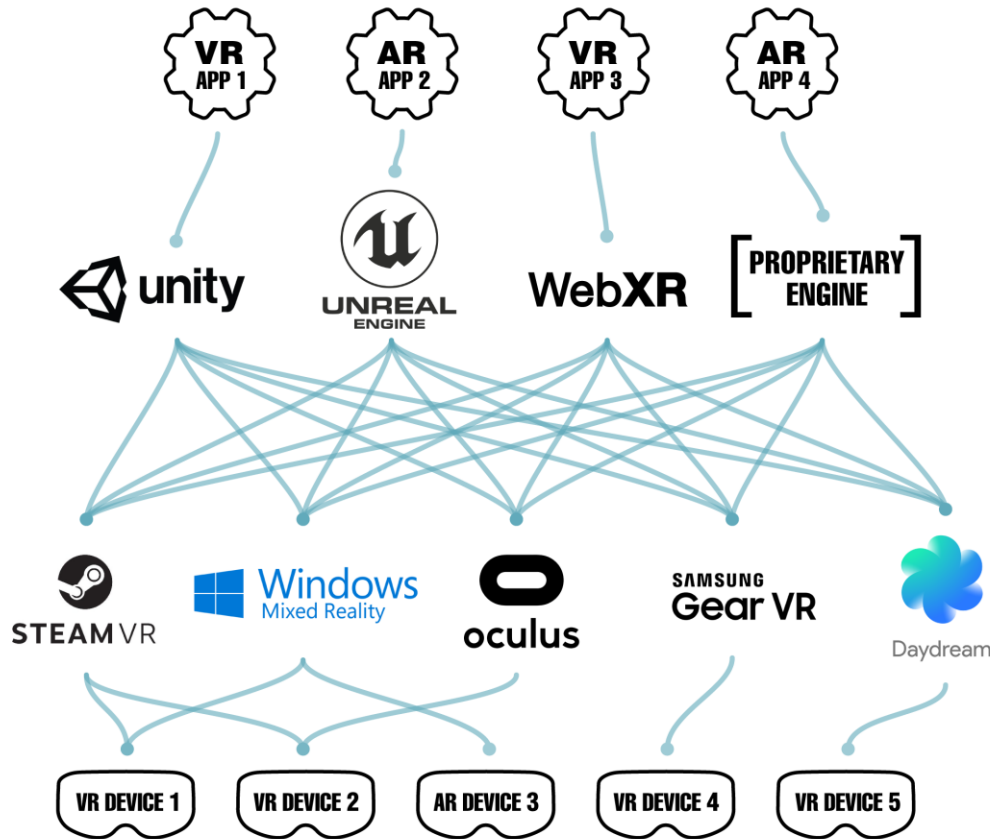
OpenXR - Solving XR Fragmentation



Before OpenXR

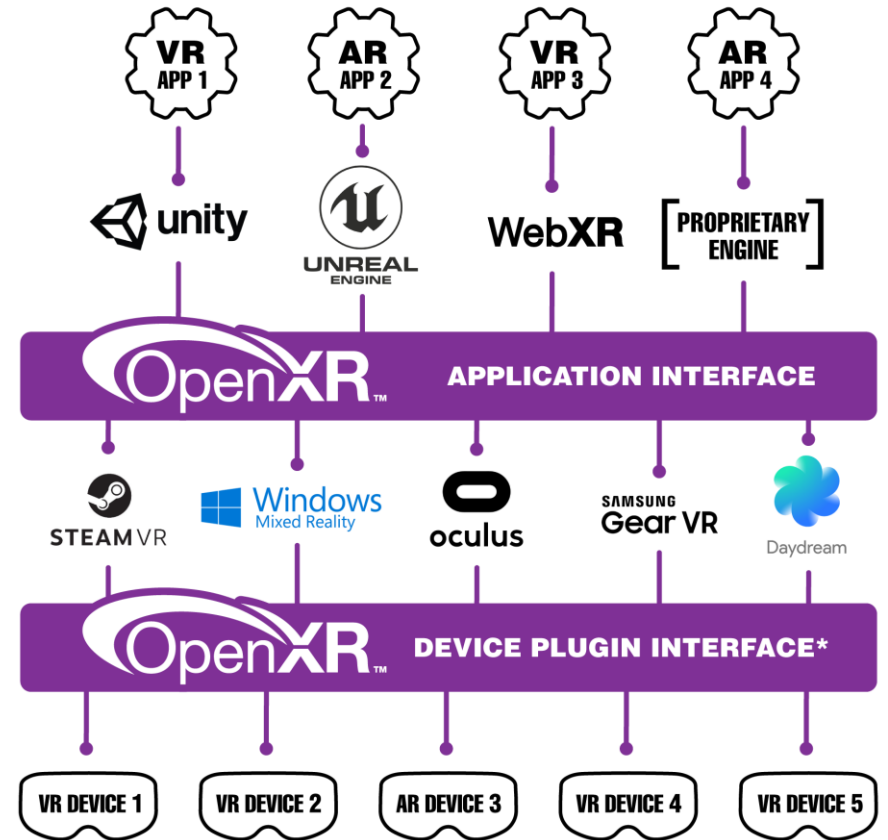
XR Market Fragmentation

OpenXR - Solving XR Fragmentation



Before OpenXR

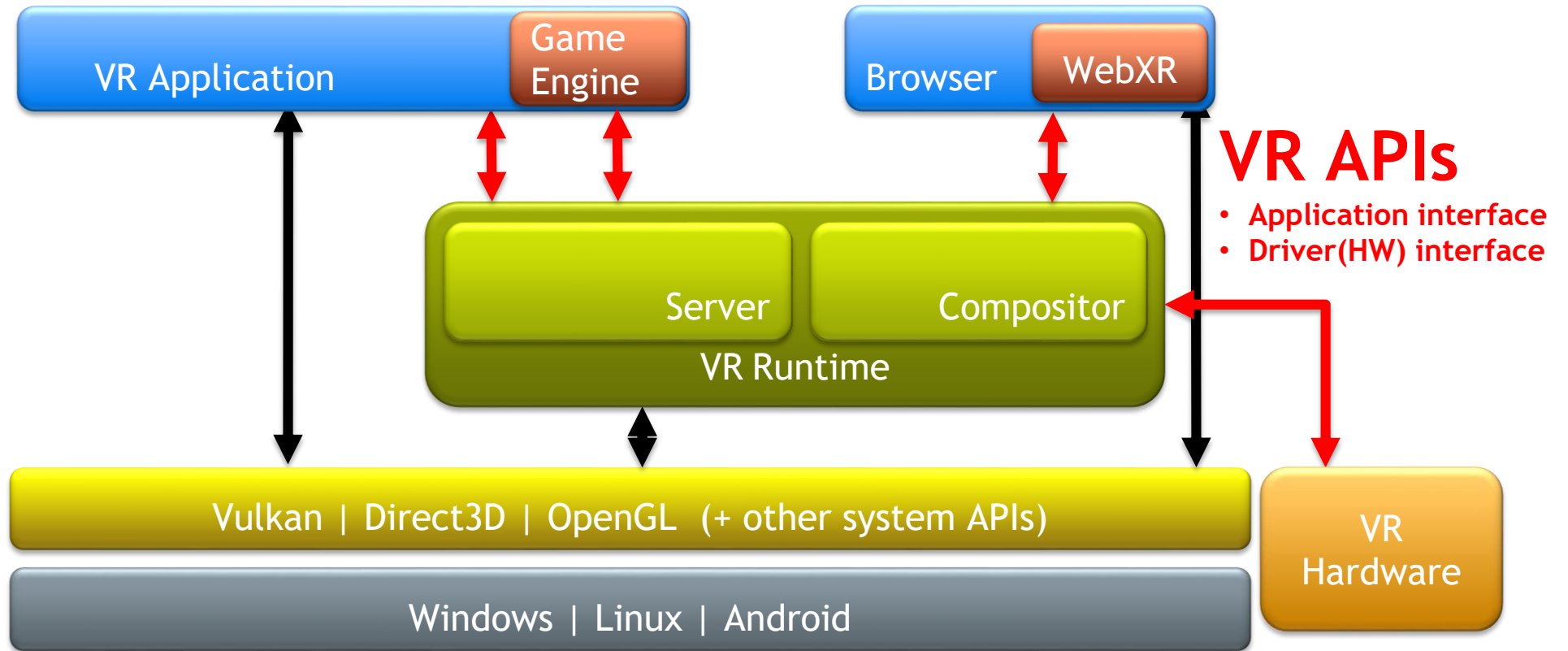
XR Market Fragmentation



After OpenXR

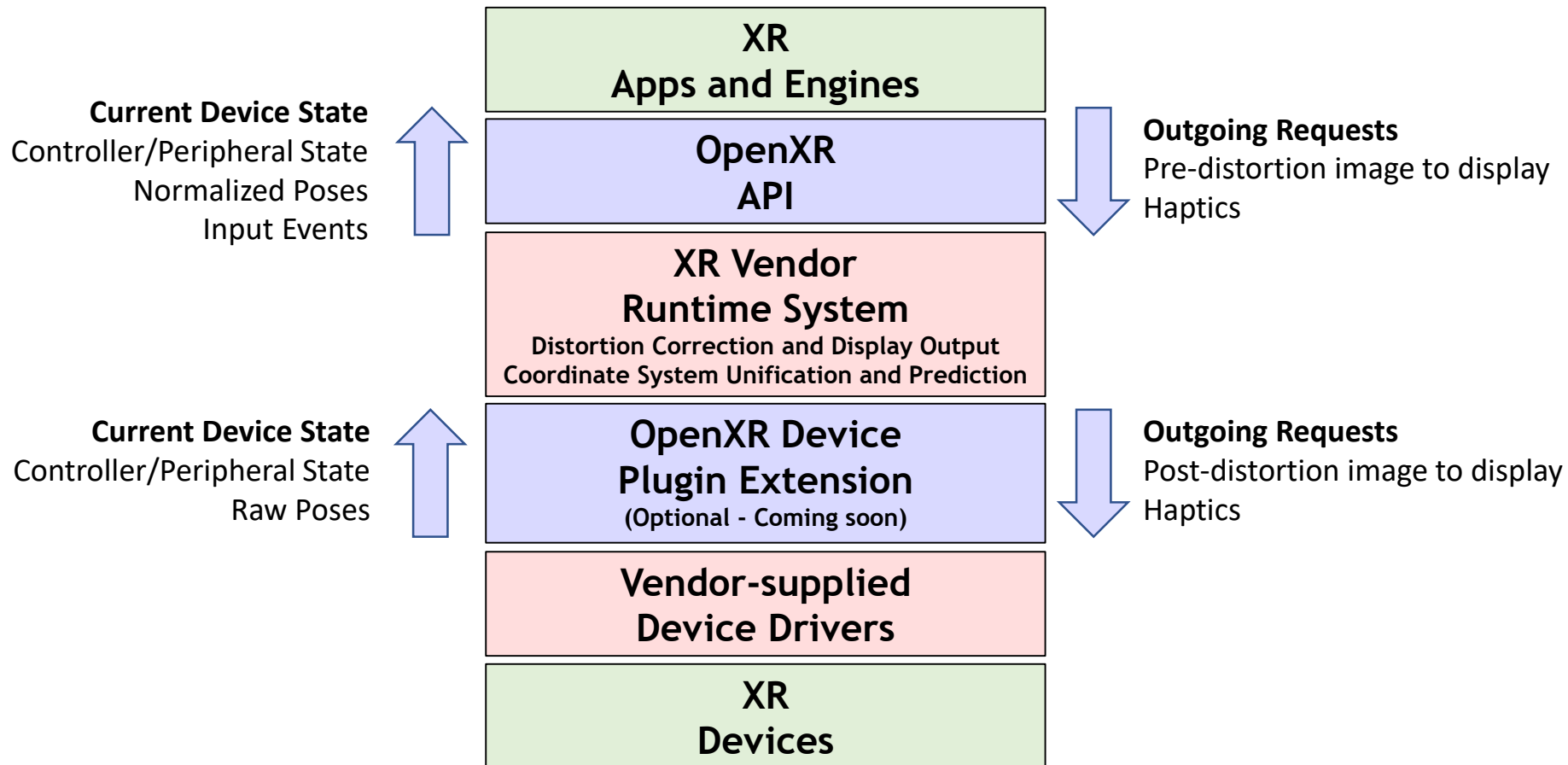
Wide interoperability of XR apps and devices

VR Software Stack (Example)

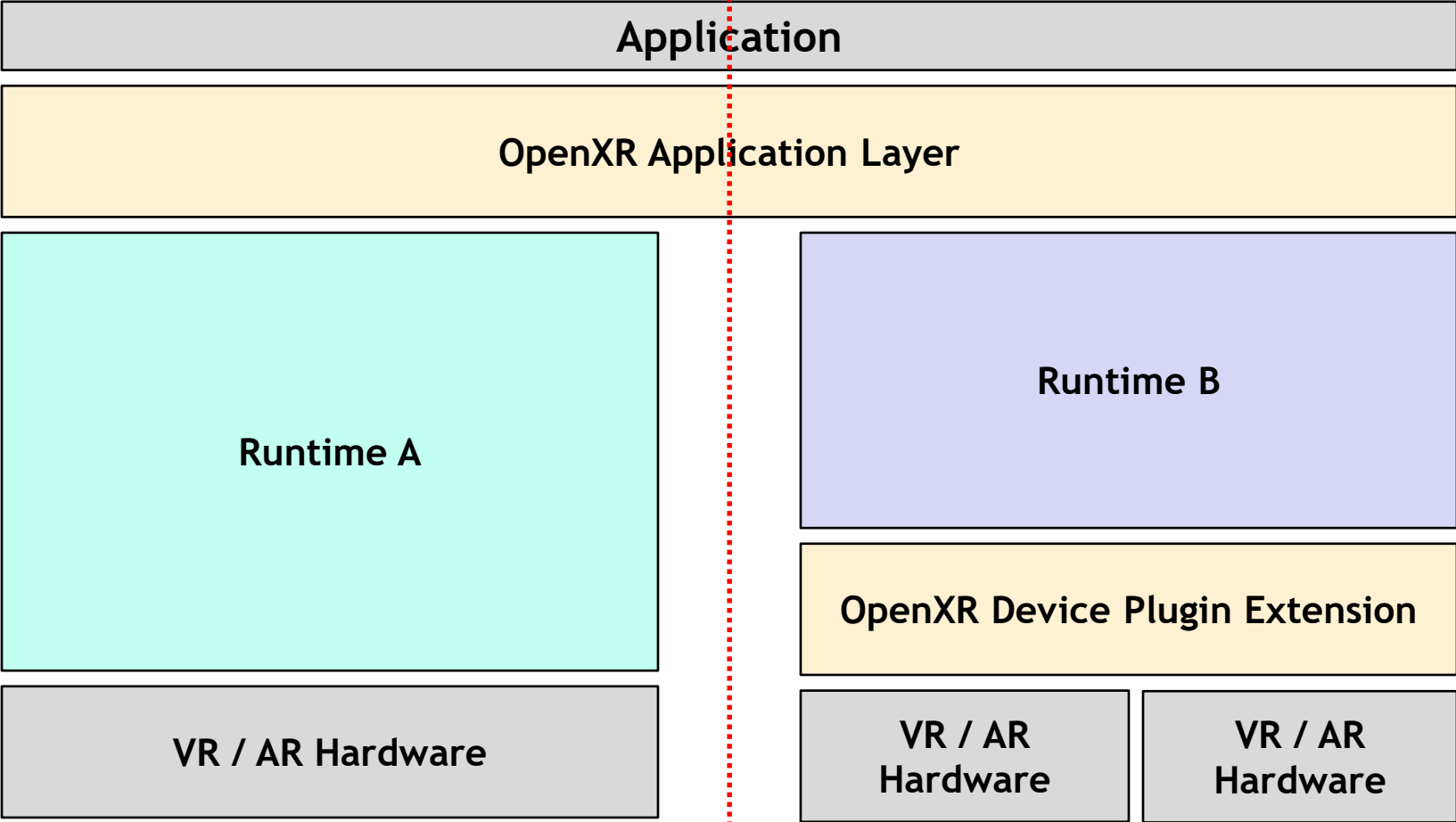


OpenXR Architecture

OpenXR does not replace XR Runtime Systems!
It enables any XR Runtime to expose CROSS-VENDOR APIs to access their functionality



The Structure



OpenXR Philosophies

1

Enable both VR and AR applications

The OpenXR standard will unify common VR and AR functionality to streamline software and hardware development for a wide variety of products and platforms

Be future-proof

2

While OpenXR 1.0 is focused on enabling the current state-of-the-art, the standard is built around a flexible architecture and extensibility to support rapid innovation in the software and hardware spaces for years to come

Do not try to predict the future of XR technology

3

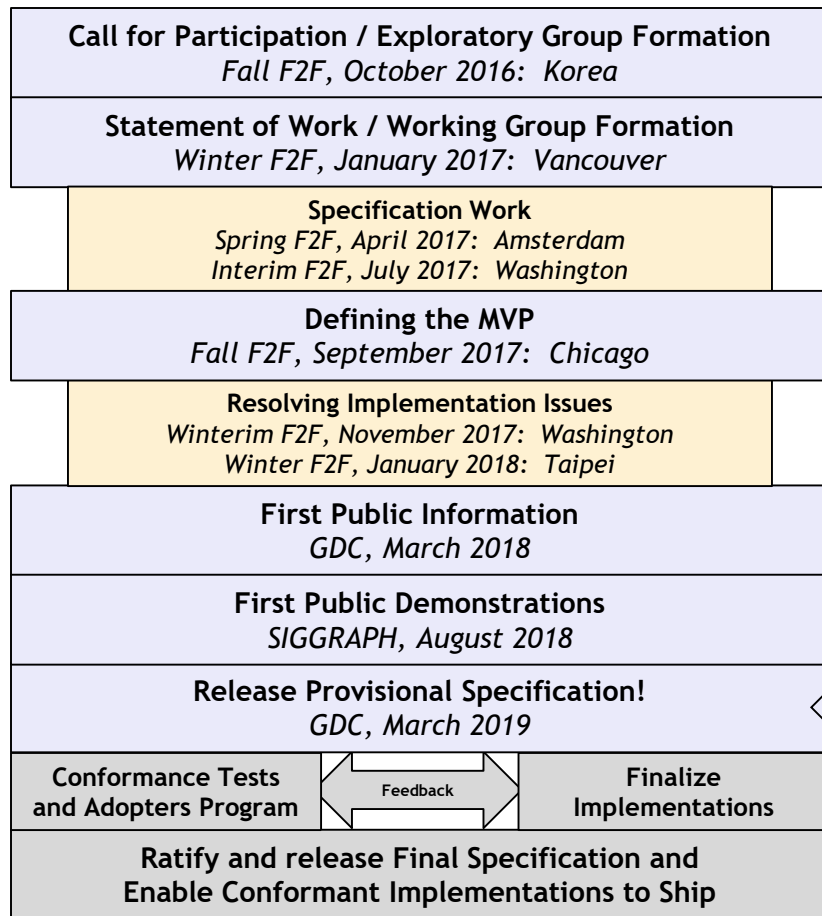
While trying to predict the future details of XR would be foolhardy, OpenXR uses forward-looking API design techniques to enable engineers to easily harness new and emerging technologies

4

Unify performance-critical concepts in XR application development

Developers can optimize to a single, predictable, universal target rather than add application complexity to handle a variety of target platforms

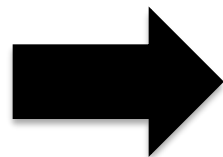
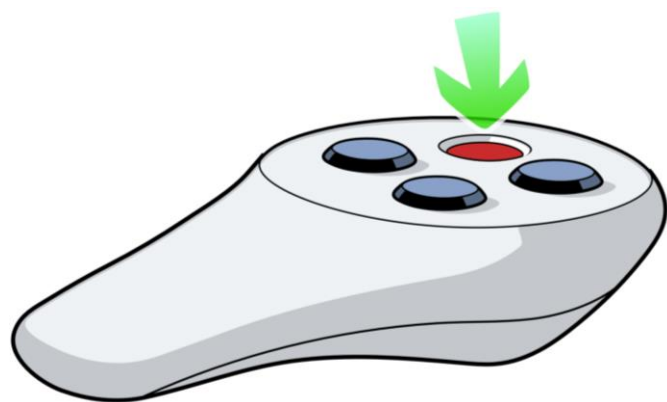
Where are we on the timeline?



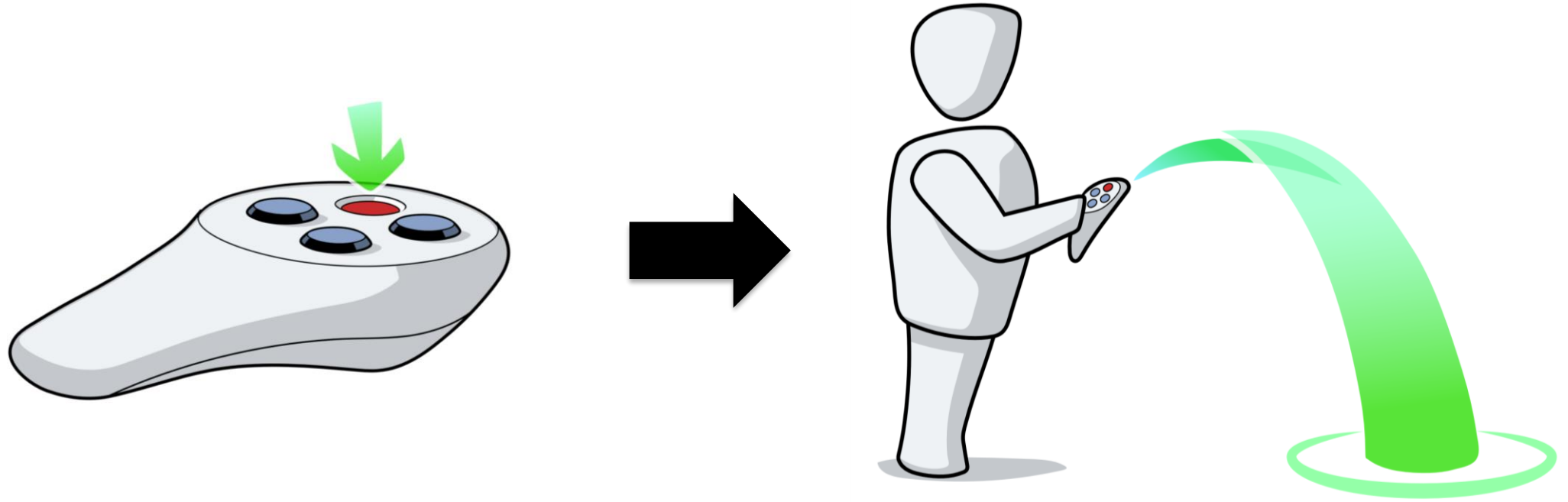
OpenXR 0.90 Provisional Specification Released
Enables industry review and feedback
First prototype implementations available

One Technical Section...

Input and Haptics



Input and Haptics

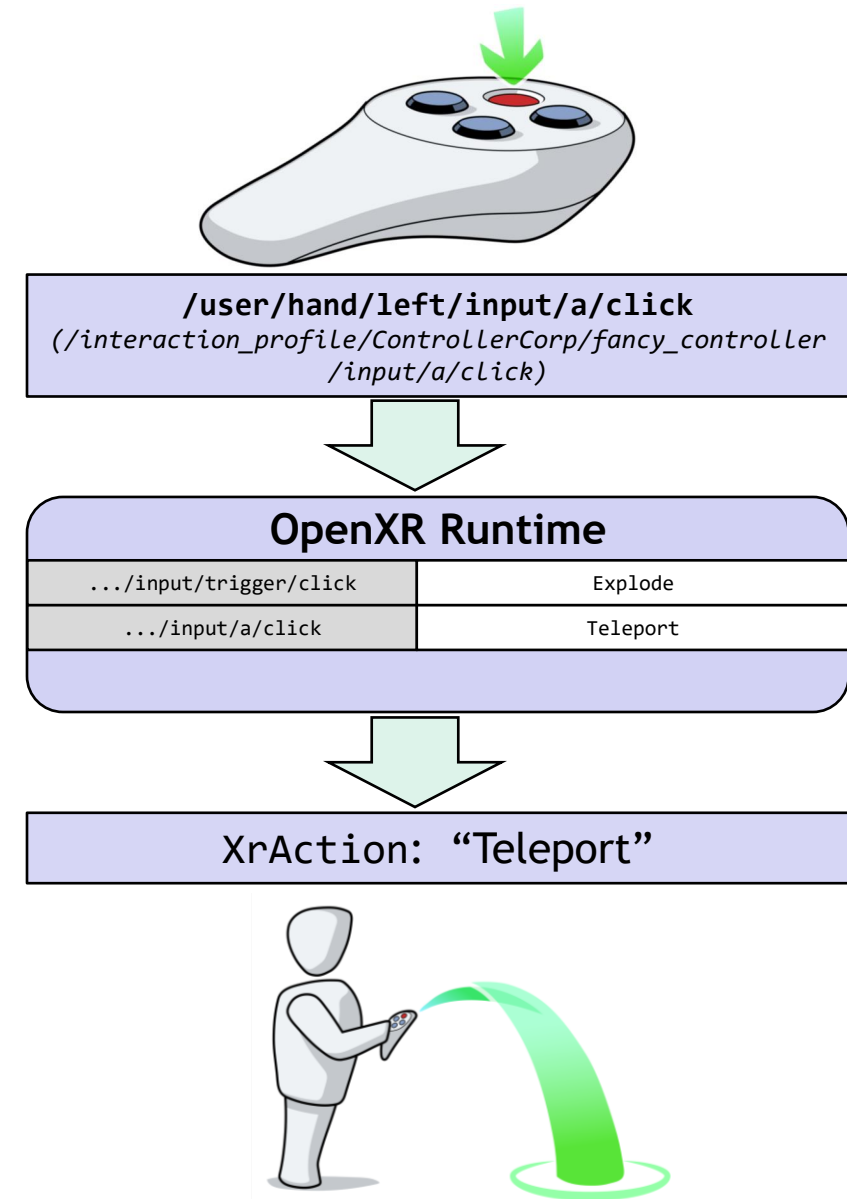


When user clicks button “a” it results in the user teleporting

Input and Haptics

Input in OpenXR goes through a layer of abstraction built around Input Actions (XrActions). These allow application developers to define input based on resulting action (e.g. “Grab”, “Jump,” “Teleport”) rather than explicitly binding controls

While the application can suggest recommended bindings, it is ultimately up to the runtime to bind input sources to actions as it sees fit (application’s recommendation, user settings in the runtime’s UI, etc)



Input and Haptics - Interaction Profiles

- Collections of input and output sources on physical devices
- Runtimes can support multiple interaction profiles

ControllerCorp's Fancy_Controller:

- /user/hand/left
- /user/hand/right

- /input/a/click
- /input/b/click
- /input/c/click
- /input/d/click
- /input/trigger/click
- /input/trigger/touch
- /input/trigger/value
- /output/haptic



example

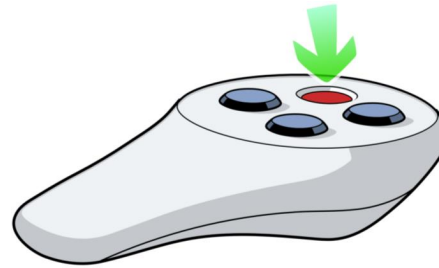
Input and Haptics - Predefined Interaction Profiles

- Interaction profiles for many current products are predefined in the OpenXR specification including:
 - Google Daydream* controller
 - HTC Vive and Vive Pro* controllers
 - Microsoft* Mixed reality motion controllers
 - Microsoft* Xbox controller
 - Oculus Go* controller
 - Oculus Touch* controllers
 - Valve Knuckles* controllers

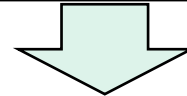
Input and Haptics - Runtime Binding Decision

- Runtime ultimately decides the bindings
 - “dev teams are ephemeral, games last forever”
 - More likely the runtime is updated than individual games
- Reasons for selecting different bindings:
 - 1. this runtime does not have ControllerCorp’s fancy_controller currently attached, but it knows how to map the inputs and outputs to the controllers that *are* attached
 - 2. Some runtimes can support user mapping of inputs such that the controls per game can be customized by the user, such as swapping trigger and button ‘a’, this enables customization without the original application knowing about it
 - 3. Some future controller is developed but the application is not updated for it, a new interaction profile can help map the actions to the new inputs
 - 4. Accessibility devices can be used and input mapped appropriately

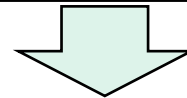
Input and Haptics



`/user/hand/left/input/a/click`
(`/interaction_profile/ControllerCorp/fancy_controller`
`/input/a/click`)



OpenXR Runtime	
<code>.../input/trigger/click</code>	Explode
<code>.../input/a/click</code>	Teleport



XrAction: "Teleport"



Extensions

Core Standard

Core concepts that are fundamental to the specification for all use cases

Examples: Instance management, tracking, frame timing

KHR Extensions

Functionality that a large class of runtimes will likely implement

Examples: Platform support , Graphic API Extensions, Device Plugin, Headless, Tracking Bounds

EXT Extensions

Functionality that a few runtimes might implement

Examples: Performance Settings, Thermals, Debug Utils

Vendor Extensions

Functionality that is limited to a specific vendor

Examples: Device specific functionality

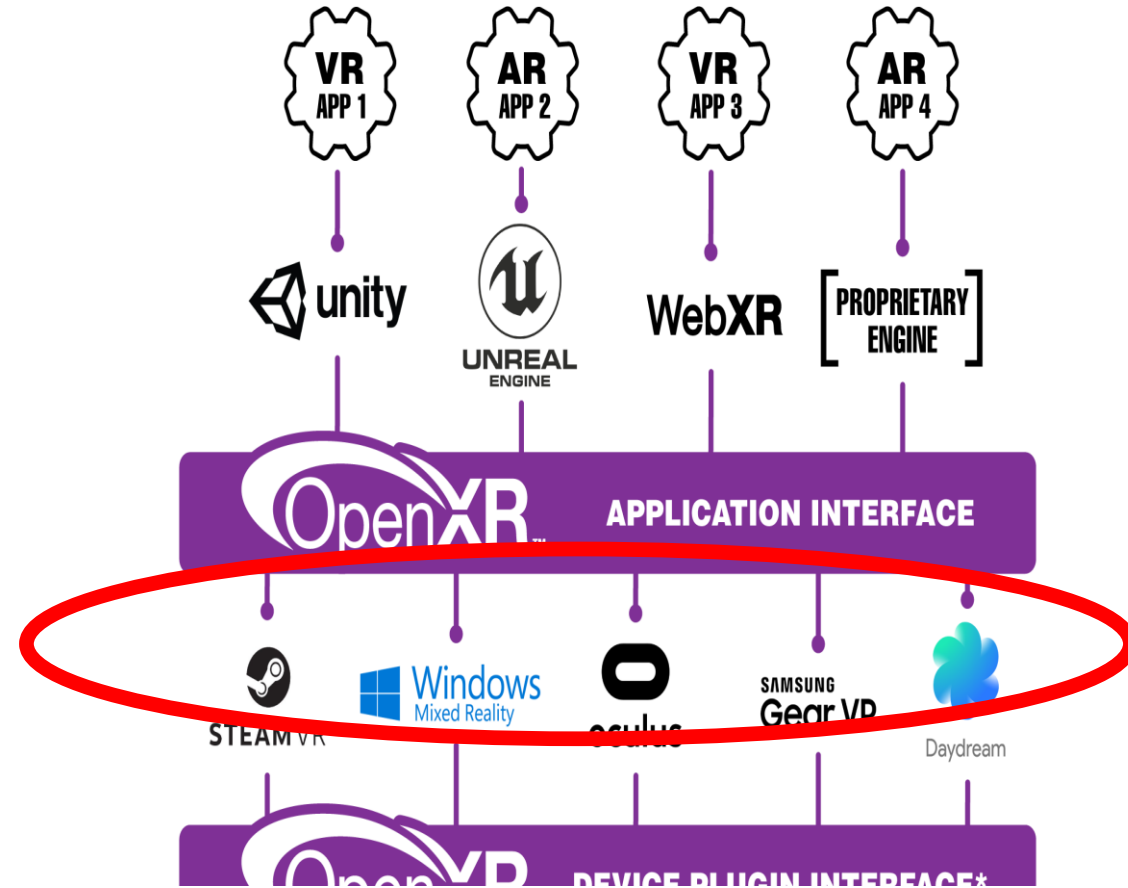
Current Provisional Extensions

- **Platform-specific support:**
 - KHR_android_create_instance
 - KHR_android_surface_swapchain
 - KHR_android_thread_settings
- **Graphics API support:**
 - KHR_D3D10_enable
 - KHR_D3D11_enable
 - KHR_D3D12_enable
 - KHR_opengl_enable
 - KHR_opengl_es_enable
 - KHR_vulkan_enable
 - KHR_vulkan_swapchain_format_list
- **Support for specific XR layer types:**
 - KHR_composition_layer_cube
 - KHR_composition_layer_depth
 - KHR_composition_layer_equirect
- **Performance improvement by masking non-visible portions of the display:**
 - KHR_visibility_mask
- **Non-display uses of OpenXR (for tracking or input-only use cases):**
 - KHR_headless
- **Time Conversion functions:**
 - KHR_convert_timespec_time
 - KHR_win32_convert_performance_counter_time

What hasn't made it in?

What hasn't made it in?

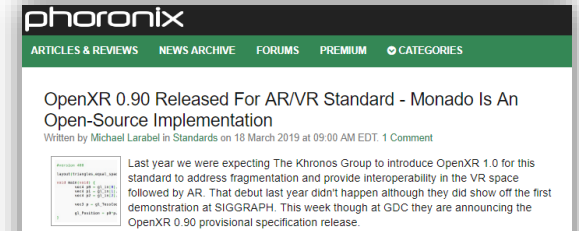
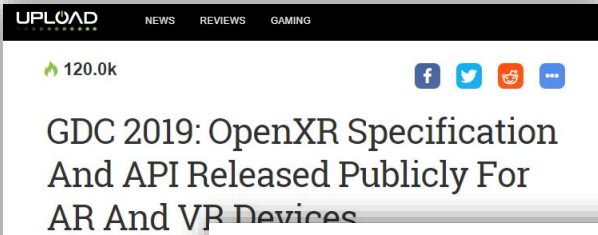
- Top priority: solve application fragmentation



There are a list of things to consider for after 1.0 or for extensions

- Many of the items on the list are obvious next steps in the progress of AR and VR development
- Won't list them here 😊

OpenXR Provisional 0.90 Release is Here!

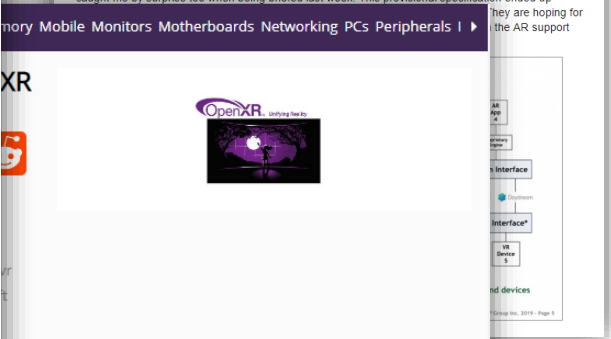
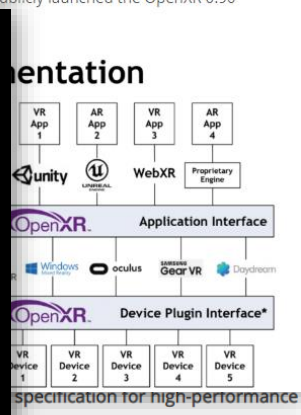


Khronos Group releases early OpenXR spec for AR and VR hardware standards

DEAN TAKAHASHI @DEANTAK



Above: Khronos Group has all the... Image Credit: Khronos Group



<https://www.roadtovr.com/openxr-0-9-provisional-release-microsoft-oculus-collabora-implementations/>

<https://uploadvr.com/openxr-provisional-release/>

https://www.phoronix.com/scan.php?page=news_item&px=Khronos-OpenXR-0.90

<https://venturebeat.com/2019/03/18/khronos-group-releases-early-openxr-spec-for-ar-and-vr-hardware-standards/>

<https://www.vrfocus.com/2019/03/khronos-openxr-0-90-provisional-spec-for-vr-ar-devices-launched-at-gdc-2019/>

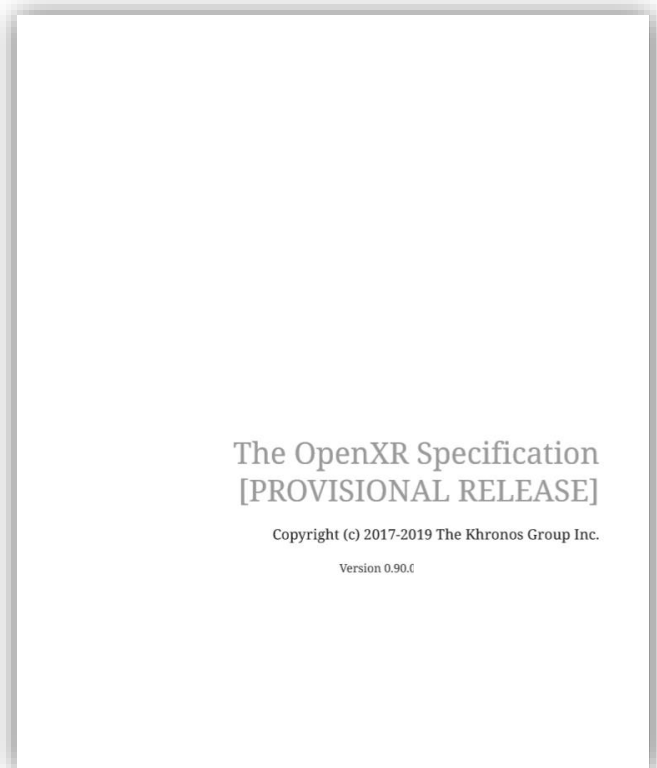
<https://bit-tech.net/news/tech/software/khronos-group-launches-openxr-090/1/>

known as the Khronos VR initiative, OpenXR was officially unveiled back in March 2017 with members including AMD, Intel, Arm, and Nvidia alongside game engine makers Epic Games and Unity. The freshly-founded working group aimed at solving the fragmentation problem in the virtual and augmented reality field, not simply by adding yet another competing standard to the pile but by promising to provide an open application interface which would allow any game engine to talk to any VR or AR platform and vice-versa.

Today's release of the OpenXR 0.90 provisional specification, announced during the Game Developer Conference, is in line with recent changes to data protection legislation in the UK and Europe we would like to direct you to our updated Privacy Policy here.

What Resources Are Available?

What Resources Are Available?



200+ page specification

What Resources Are Available?

The OpenXR Specification [PROVISIONAL RELEASE]

Copyright (c) 2017-2019 The Khronos Group Inc.

Version 0.90.0

200+ page specification

xrCreateInstance(3) Manual Page

NAME

xrCreateInstance - Creates an OpenXR Instance

C Specification

The `xrCreateInstance` function is defined as:

```
XrResult xrCreateInstance(  
    const XrInstanceCreateInfo* createInfo,  
    XrInstance* instance);
```

Parameters

Parameter Descriptions

- `createInfo` points to an instance of `XrInstanceCreateInfo` controlling creation of the instance.
- `instance` points to an `XrInstance` handle in which the resulting instance is returned.

Description

`xrCreateInstance` creates the `XrInstance`, then enables and initializes global API layers and extensions requested by the application. If an extension is provided by an API layer, both the API layer and extension **must** be specified at `xrCreateInstance` time. If a specified API layer cannot be found, no `XrInstance` will be created and the function will return `XR_ERROR_API_LAYER_NOT_PRESENT`. Likewise, if a specified extension cannot be found the call will return `XR_ERROR_EXTENSION_NOT_PRESENT` and no `XrInstance` will be created. Additionally, some runtimes may limit the number of concurrent instances that may be in use. If the application attempts to create more instances than a runtime can simultaneously support, `xrCreateInstance` will return `XR_ERROR_LIMIT_REACHED`.

If the `XrInstanceCreateInfo` struct contains a platform-specific extension for a platform other than the target platform, `XR_ERROR_INITIALIZATION_FAILED` will be returned. The same is true if a mandatory platform-specific extension is defined for the target platform but no matching extension struct is provided in `XrInstanceCreateInfo`.

Valid Usage (Implicit)

- `createInfo` **must** be a pointer to a valid `XrInstanceCreateInfo` structure
- `instance` **must** be a pointer to an `XrInstance` handle

Reference Pages

What Resources Are Available?

The OpenXR Specification
[PROVISIONAL RELEASE]

Copyright (c) 2017-2019 The Khronos Group Inc.

Version 0.90.0

200+ page specification

xrCreateInstance(3) Manual Page

NAME

xrCreateInstance - Creates an OpenXR Instance

C Specification

The `xrCreateInstance` function is defined as:

```
XrResult xrCreateInstance(  
    const XrInstanceCreateInfo* createInfo,  
    XrInstance* instance);
```

Parameters

Parameter Descriptions

- `createInfo` points to an instance of `XrInstanceCreateInfo` controlling creation of the instance.
- `instance` points to an `XrInstance` handle in which the resulting instance is returned.

Description

`xrCreateInstance` creates the `XrInstance`, then enables and initializes global API layers and extensions requested by the application. If an extension is provided by an API layer, both the API layer and extension must be specified at `xrCreateInstance` time. If a specified API layer cannot be found, no `XrInstance` will be created and the function will return `XR_ERROR_API_LAYER_NOT_PRESENT`. Likewise, if a specified extension cannot be found the call will return `XR_ERROR_EXTENSION_NOT_PRESENT` and no `XrInstance` will be created. Additionally, some runtimes may limit the number of concurrent instances that may be in use. If the application attempts to create more instances than a runtime can simultaneously support, `xrCreateInstance` will return `XR_ERROR_LIMIT_REACHED`.

If the `XrInstanceCreateInfo` struct contains a platform-specific extension for a platform other than the target platform, `XR_ERROR_INITIALIZATION_FAILED` will be returned. The same is true if a mandatory platform-specific extension is defined for the target platform but no matching extension struct is provided in `XrInstanceCreateInfo`.

Valid Usage (Implicit)

- `createInfo` must be a pointer to a valid `XrInstanceCreateInfo` structure
- `instance` must be a pointer to an `XrInstance` handle

The diagram illustrates the OpenXR architecture, showing the flow from API Layers to various runtime components. It is organized into several sections:

- API Layers:** Includes `EnumerateApiLayerProperties`.
- Instances:** Includes `GetInstanceProcAddr`, `EnumerateInstanceExtensionProperties`, `CreateInstance`, `DestroyInstance`, and `GetInstanceProperties`.
- Events:** Includes `PollEvent`.
- Sessions:** Includes `CreateSession`, `EndSession`, `BeginSession`, and `DestroySession`.
- Swapchains:** Includes `EnumerateSwapchainFormats`, `CreateSwapchain`, `DestroySwapchain`, `EnumerateSwapchainImages`, `AcquireSwapchainImage`, `WaitSwapchainImage`, and `ReleaseSwapchainImage`.
- Spaces:** Includes `EnumerateReferenceSpaces`, `CreateReferenceSpace`, `CreateActionSpace`, `LocateSpace`, `GetReferenceSpaceBoundsRect`, and `DestroySpace`.
- Helper Functions:** Includes `ResultToString`, `StructureTypeToString`, `StringToPath`, and `PathToString`.
- Systems:** Includes `GetSystem`, `GetSystemProperties`, `EnumerateSystemRenderModes`, and `EnumerateSystemViewConfigurations`.
- View Configurations:** Includes `EnumerateViewConfigurations`, `GetViewConfigurationProperties`, `EnumerateViewConfigurationViews`, and `LocateViews`.
- Input and Haptics:** Includes `CreateAction`, `DestroyAction`, `CreateActionSet`, `DestroyActionSet`, `GetActionBase*`, `SyncActionData`, `GetSoundSourceForAction`, `GetInputSourceLocationName`, `SetInteractionProfile/aggregatedbindings`, `GetCurrentInteractionProfile`, `ApplyHapticFeedback`, and `DestroyHapticFeedback`.
- Render Loop:** Includes `WaitFrame`, `BeginFrame`, and `EndFrame`.
- Helper Functions:** Includes `ResultToString`, `StructureTypeToString`, `StringToPath`, and `PathToString`.
- Input and Haptics (Detailed):** Shows the process of creating action and action spaces, setting up interaction profile bindings, and getting action states. It includes code snippets for `xrCreateAction`, `xrCreateActionSpace`, `xrSetInteractionProfile/aggregatedbindings`, and `xrGetActionStateBoolean`.

Overview Guide

What Resources Are Available?

- <https://github.com/KhronosGroup/OpenXR-Docs>
- Contains the source for generating the specification document and reference pages, scripts to be added soon
- Contains the openxr header files

OpenXR[®] API Documentation Project

[NOTE: This is the initial set up for the provisional release of the specification. Not all the files are populated yet, and expect significant changes as the spec moves towards 1.0.]

This repository contains sources for the formal documentation of the OpenXR API. This includes:

- the OpenXR API Specification
- OpenXR header files
- related tools and scripts.

The authoritative public repository is located at <https://github.com/KhronosGroup/OpenXR-Docs/>. It hosts public Issue tracker, and accepts patches (Pull Requests) from the general public.

Directory Structure

The directory structure is as follows:

README.adoc	This file
COPYING.md	Copyright and licensing information
CODE_OF_CONDUCT.md	Code of Conduct
specification/	Specification - files to generate the spec
include/openxr/	OpenXR headers, generated from the Registry

What Resources Are Available?

- <https://github.com/KhronosGroup/OpenXR-Registry>
- Contains the specification, reference pages, and overview guide

OpenXR-Registry

The OpenXR-Registry repository contains the OpenXR API and Extension Registry, including generated specifications and reference pages, and reference cards. The sources for these documents are mostly found in the separate <https://github.com/KhronosGroup/OpenXR-Docs> repository; this repository is used as a backing store for the web view of the registry at <https://www.khronos.org/registry/openxr/>. Commits to the master branch of OpenXR-Registry will be reflected in the web view.

Interesting files in this repository include:

- index.php - toplevel index page for the web view. This relies on PHP include files found elsewhere on www.khronos.org and so is not very useful in isolation.
- specs/0.90/ - OpenXR 0.90 Provisional API specifications and reference pages and API reference card.

What Resources Are Available?

- <https://github.com/KhronosGroup/OpenXR-SDK>
- Contains the source for:
 - Loader
 - Some basic API layers
 - Test sample
- For the current best example code, see: `src/tests/hello_xr`

OpenXR[®] Software Development Kit (SDK) Project

[NOTE: This repository contains components that may eventually be assembled into an SDK, but are currently not being packaged into a distributable SDK.]

This repository contains source code and build scripts for implementations of the OpenXR loader, validation layers, and code samples.

The authoritative public repository is located at <https://github.com/KhronosGroup/OpenXR-SDK/>. It hosts public Issue tracker, and accepts patches (Pull Requests) from the general public.

Directory Structure

BUILDING.md	Instructions for building the projects
README.md	This file
COPYING.md	Copyright and licensing information
CODE_OF_CONDUCT.md	Code of Conduct
external/	External code for projects in the repo
include/	OpenXR platform include file
specification/	xr.xml file
src/	Source code for various projects
src/api_layer	Sample code for developing API layers
src/loader	OpenXR loader code
src/tests/	various test code (if looking for sample code start with hello_xr/)

Currently the best sample code is in `src/tests/hello_xr/`. More will be added in the future.

This structure is for the provisional specification. Things are incomplete at launch but will be added to going forward.

Additional Resources

- **OpenXR Landing Page - Specification, Reference Pages, Sample Code, Overview**
 - <https://www.khronos.org/openxr>
- **OpenXR Forum and Slack Channel**
 - Forum: <https://khr.io/openxrfeedback>
 - Discussion: <https://khr.io/slack>
- **Detailed specification overview and SIGGRAPH session videos**
 - <https://www.khronos.org/developers/library/2018-siggraph>
- **Vendor prototype runtime implementations**
 - Collabora: open source implementation
<http://monado.dev>
 - Microsoft: OpenXR runtime for Windows Mixed Reality headsets
<https://aka.ms/openxr>
- **Khronos GDC Sessions - including OpenXR Presentation and demos**
 - <https://www.khronos.org/events/2019-gdc>

Engine and Platform Support

Vinay Narayan, vice president, platform strategy, HTC

“HTC VIVE is committed to creating a viable ecosystem for the XR industry which is why we are proud to support OpenXR. Bringing the community together to help define standards and best practices, allows all of us to move forward, together.”



Tim Sweeney, founder and CEO of Epic Games

*“Epic believes that open standards like OpenXR are essential foundations for a vibrant, multi-platform VR and AR industry in the coming years. **Epic plans to continue supporting OpenXR in Unreal Engine 4.**”*

Nate Mitchell, Oculus Co-founder and head of VR product, Facebook

*“Facebook and Oculus continue to believe in the value the OpenXR standard delivers to users and developers. **We plan to provide runtime support for apps built on OpenXR 1.0 on the Rift and Quest platforms later this year.**”*



Alex Kipman, technical fellow, Microsoft

*“Microsoft believes that for mixed reality to thrive, it must be open for everyone: open stores, open browsers and open developer platforms. **We're dedicated to supporting the launch of OpenXR this year on Windows Mixed Reality and HoloLens 2. To help developers provide feedback, we're releasing today a developer preview of our OpenXR runtime with support for Windows Mixed Reality headsets.**”*



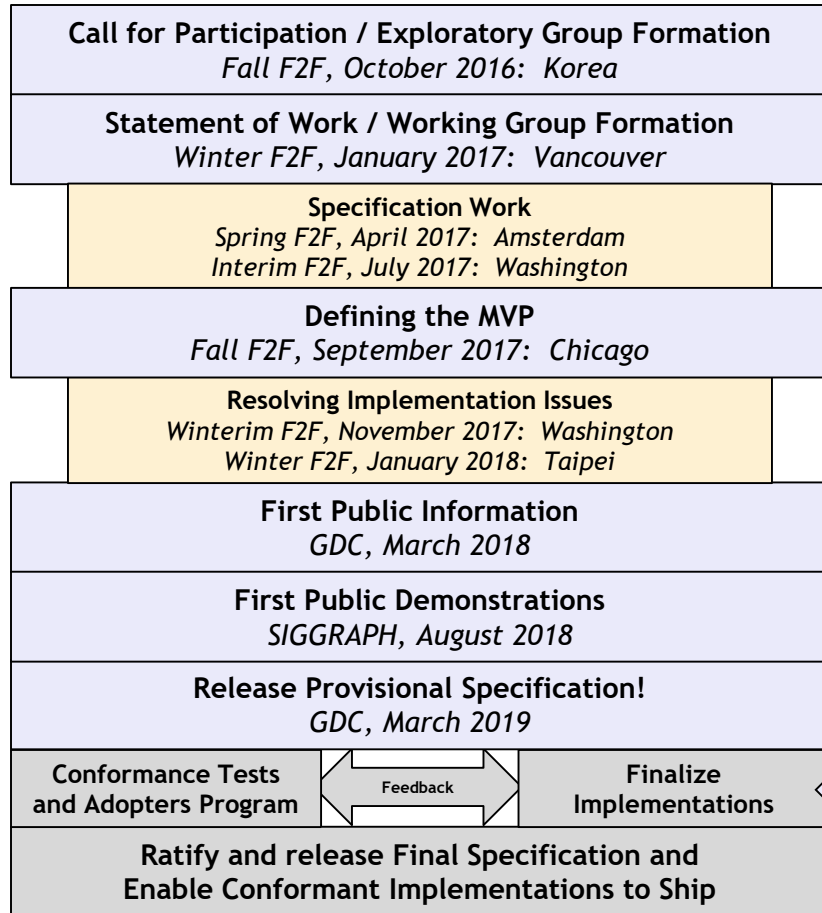
Philippe Kalaf, CEO, Collabora

*Collabora is excited to announce **Monado**, an open source implementation of the newly released **OpenXR spec**. More than just a vendor SDK, Monado is an open source project and codebase to harness and focus wider community effort around XR technologies.*



What's Next?

What's Next?



Drive towards 1.0 release!

What's Next?

- **Continue Refining the Specification**
 - Very unlikely any new functionality between now and 1.0
- **Incorporate Community Feedback**
 - Bug fixes, things missing, etc.
- **Establish an Adopter's Program**
- **Develop conformance tests!**
 - Major next hurdle
 - Conformance tests are critical to the health of an API (particularly a new one)
 - Potentially lots of corner cases requiring tests
 - Make sure not just the common paths work well
 - Make sure a particular vendor's implementation becomes the de facto standards
 - How to test AR and VR systems end to end ?
 - Can you test successfully without robotic arms and cameras?

Thanks!

- To these companies for enabling their engineers to dedicate time to OpenXR!



Thanks to the Engineers!

Adam Gousetis, Google | Alex Turner, Microsoft | Andreas Loeve Selvik, Arm | Andres Rodriguez, Valve Software | Armelle Laine, Qualcomm Technologies, Inc | Blake Taylor, Magic Leap | Brad Grantham, Google | Brandon Jones, Google | Brent E. Insko, Intel | Brent Wilson, Microsoft | Bryce Hutchings, Microsoft | Cass Everitt, Facebook | Charles Egenbacher, Epic Games | Christoph Haag, Collabora | Craig Donner, Google | Dan Ginsburg, Valve Software | Dave Houlton, LunarG | Dave Shreiner, Unity Technologies | Denny Rönngren, Tobii | Dmitriy Vasilev, Samsung | Doug Twileager, ZSpace | Ed Hutchins, Facebook | Gloria Kennickell, Facebook | Gregory Greeby, AMD | Guodong Chen, Huawei | Jakob Bornecrantz, Collabora | Jared Cheshier, PlutoVR | Javier Martinez, Intel | Jeff Bellinghausen, Valve Software | Jiehua Guo, Huawei | Joe Ludwig, Valve Software | Johannes van Waveren, Facebook | Jon Leech, Khronos | Jonathan Wright, Facebook | Juan Wee, Samsung | Jules Blok, Epic Games | Karl Schultz, LunarG | Kaye Mason, Google | Krzysztof Kosiński, Google | Lachlan Ford, Microsoft | Lubosz Sarnecki, Collabora | Mark Young, LunarG | Martin Renschler, Qualcomm Technologies, Inc. | Matias Koskela, Tampere University of Technology | Matt Wash, Arm | Mattias Brand, Tobii | Mattias O. Karlsson, Tobii | Michael Gatson, Dell | Minmin Gong, Microsoft | Mitch Singer, AMD | Nell Waliczek, Microsoft | Nick Whiting, Epic Games | Nigel Williams, Sony | Paul Pedriana, Facebook | Peter Kuhn, Unity Technologies | Peter Peterson, HP Inc. | Pierre-Loup Griffais, Valve Software | Rajeev Gupta, Sony | Remi Arnaud, Starbreeze | Remy Zimmerman, Logitech | River Gillis, Google | Robert Memmott, Facebook | Robert Menzel, NVIDIA | Robert Simpson, Qualcomm Technologies, Inc. | Robin Bourianes, Starbreeze | Ryan Pavlik, Collabora | Ryan Vance, Epic Games | Sam Martin, Arm | Satish Salian, NVIDIA | Scott Flynn, Unity Technologies | Sophia Baldonado, PlutoVR | Sungye Kim, Intel | Tom Flynn, Samsung | Trevor F. Smith, Mozilla | Vivek Viswanathan, Dell | Yin Li, Microsoft | Yuval Boger, Sensics

Recap

- What is OpenXR?
- A Brief History of the Standard
- What are the Problems we are trying to Solve
- OpenXR Timeline of Development
- Provisional Release
- What's Next?

OpenXR Win-Win-Win

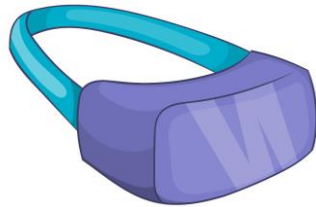
XR End-Users

Can run the apps they want on their system
- reducing market confusion and increasing consumer confidence



XR Vendors

Can bring more applications onto their platform by leveraging the OpenXR content ecosystem



XR ISVs

Can easily ship on more platforms for increased market reach

Khronos Mission

KHRONOS GROUP

Over 140 members worldwide
Any company is welcome to join

PROMOTER MEMBERS

AMD, Apple, arm, EPIC GAMES, Google, HUAWEI, Imagination, intel, NVIDIA, QUALCOMM, SAMSUNG, SONY, VALVE, VeriSilicon

3D Incorporated, Adobe, AMOTIVE, Almalence, amazon.com, Au-Zone, AXELL CORPORATION, AXIS COMMUNICATIONS, BASE MARK, BINOMIAL, BLIZZARD, BOEING, BROADCOM, BRENWILL, cadence, CAICT, codeplay, codeweavers, Continental, COREAVI, DASSAULT SYSTEMES, DELL, DGG, CATAPULT Digital, DisplayLink, OMP, Esperanto Technologies, ETRI, EYEWAY, Fraunhofer, FUTURMARK CORPORATION, Gaijin, hp, HITACHI, htc, igalia, IKEA Communications AB, Imperial College London, ITRI, KALRAY, KDAB, KNU, LG, Linaro, logitech, LUNAR, magic leap, matrox, MEDIATEK, mercury systems, Microsoft, MIT Lincoln Laboratory, MITSUBISHI ELECTRIC, mobica, mozilla, NEC, NIHON UNIVERSITY, Nintendo, NOKIA, NSI-TEXE, NXP, oppo, OSU, otoy, OXIDE, PEAKHILLSGROUP, Pico, pluto, RAZER, RENESAS, Rockwell Collins, 서울대학교, 兆芯, SIRU, socionext, STARVR, STREAM, SYNOPSIS, TAKUMI, TAMPERE UNIVERSITY OF TECHNOLOGY, TEXAS INSTRUMENTS, thinci, Think Silicon, tobii, umbra, unity, UTAH, University of BRISTOL, UNIVERSITY OF TORONTO, University of Windsor, UX3D, VASTO, Visteon, vmware, vrg, XILINX, ZERO LATENCY, zSpace

Khronos members are **industry leaders** from around the world that join to **safely cooperate** - to advance their own businesses and the industry as a whole



Khronos is an **open**, member-driven industry consortium developing **royalty-free standards**, and vibrant ecosystems, to harness the power of **silicon acceleration** for demanding **graphics rendering** and **computationally intensive** applications

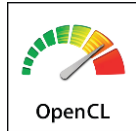
Active Khronos Open Standards

COLLADA™



3D Asset Authoring
and Delivery

Parallel Computation, Vision,
Machine Learning and Inferencing



High-performance
3D Graphics

High-performance access to AR
and VR platforms and devices



Join Khronos!

- **Get more involved**
- **Have direct impact on the direction of the API**
- **Be part of the effort to deliver OpenXR 1.0!**



Thank You!



Questions?

Backup